Parallel Port Mains Switching Interface

switch radio-controlled mains sockets on and off from a PC

by K. Heiden

The wide availability of low-cost switched mains sockets with radio remote control provides a simple way to control electrical equipment, lights, etc. using a PC. Using radio not only means that we do not need a cable, but also avoids problems with electrical safety as the interface is electrically isolated from the mains supply.

As USB interfaces become more widespread on PCs the serial and parallel ports are becoming less and less important.
Most computers, however, still offer the ‘legacy’ ports which are available for new applications, now that devices such as printers tend to use USB. The parallel port, for example, is ideally suited for controlling electrical devices remotely. Using a simple switch interface it is possible to turn household appliances on and off under PC control.

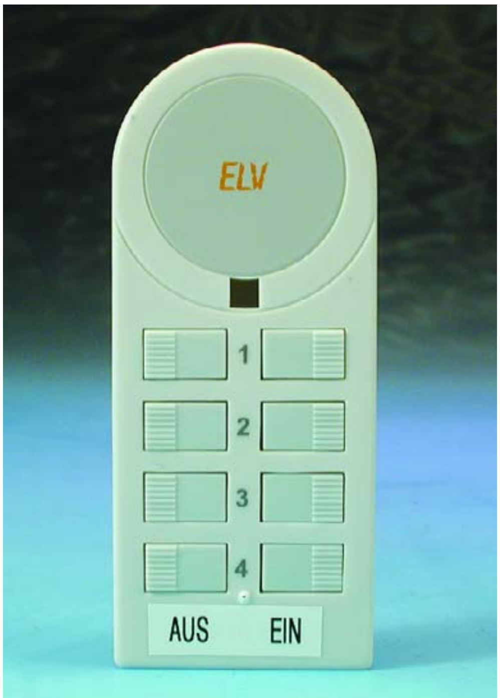

Figure 1. The handheld transmitter (ELV part code FS 10 S8-2) before modification.

So as not to add to the cable spaghetti already associated with the PC, the remote control described here uses a modified radio-controlled mains switch. These are available from various outlets including specialist electronics shops as well as DIY and housewares shops. Generally two or three switched sockets are sold as a set along with a single hand-held remote control (see **Figure 1** and **Figure 2**).

In order to control the switches from a PC we will need an interface to connect the parallel port of the PC to the hand-held remote control transmitter. If the facility for manual control must be retained, the best approach is to purchase another transmitter, as the unit connected to the interface will be attached to the parallel port using only a short cable and so it will not be convenient to operate it manually.

Interface Circuit

The parallel port interface circuit (**Figure 3**) is essentially divided into two parts: the part that drives the remote control transmitter using optocouplers, and two optional extra switching stages built around IC1 and two relays. This switching circuit allows for the control of two additional devices, located near to the PC, directly from the interface. If this feature is not required, the switching stages can be left out, and signals D0, D1 and D3 (port pins 2, 3 and 5 of the 25-way sub-D plug) left unconnected.

Port signals D4 to D7 (pins 6, 7, 8 and 9 of the 25-way sub-D plug) are connected to the remote control transmitter via a quad optocoupler device (IC2, type ILQ74). Figure 1 also shows the pinout of this device.

The circuit diagram only shows the part of the remote control transmitter circuit that is connected to the interface circuit, namely the pushbuttons and the connections for the power supply. The pushbutton connections for the transmitter are brought out on four wires labelled A to D and connected to the output transistors of the optocouplers (see

Figure 2. A radio-controlled switched mains socket (ELV FS 10 system).

also **Figure 4**). When the port pin driving an optocoupler goes high, its output transistor conducts, bridging the corresponding pushbutton contacts on the transmitter. The effect is the same as if the button had been pressed (channel 2 on/off, channel 3 on/off and so on). We can arrange for the pin on the parallel port to be taken high using suitable software. The components between the parallel port and the optocoupler (the series resistor and the diode) serve to limit current flow and protect both the input LED of the optocoupler and the output transistors of the parallel port.

The interface circuit also includes a power supply for the remote control transmitter. The transmitter batteries (two button cells in the type we used) should be removed and connections made to the points marked '+' and '–' in the circuit diagram. A voltage of 3 V for the transmitter is derived from the 12 V supply of the interface using D2, R11, C8 and Zener diode D3. The interface circuit itself can be powered from a 12 V mains adaptor connected to K1 or by connecting to the PC's internal 12 V supply.

The optional switching stages connected to port pins D0, D1 and D3 are equipped with one normal relay (with one coil) and one latching relay (with two coils). They are controlled by driver IC1 (type ULN2803).

The latching relay (RE1) is controlled using port signals D0 and D1 and has the feature that its state is preserved when power is switched off. In contrast, RE2 is an ordinary, non-latching, relay and is controlled using D3. Its contacts are closed only while current flows in the coil. Switching pulses can be generated using this relay with a duration under software control. Thus it is suitable, for example, for driving an electrically-operated door latch from the PC, connecting the relay contacts in parallel with the button for opening the door.

You may be wondering why there are no protection diodes across the relay coils: this is because suitable diodes are already built into the ULN2803.

A 25-way male sub-D connector is required to connect the interface circuit to the parallel port. A cable with at least eight conductors is required for the connection, with the 25-way sub-D connector at one and suitable connector at the other. If it is desired to use a spare ordinary printer cable, a 36-way Centronics-style socket will be required.

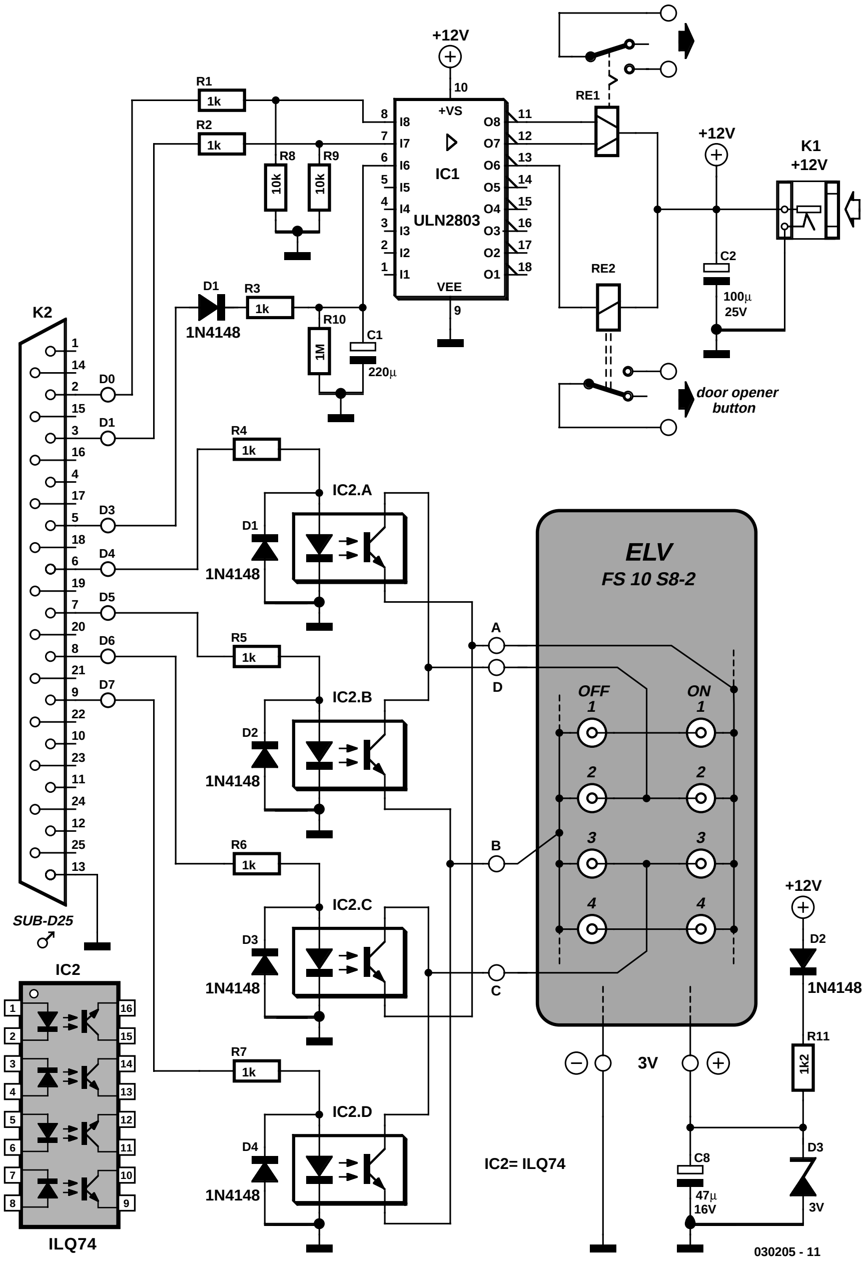

Figure 3. The circuit of the parallel port interface connects to the handheld transmitter via optocouplers. Two relays are also available.

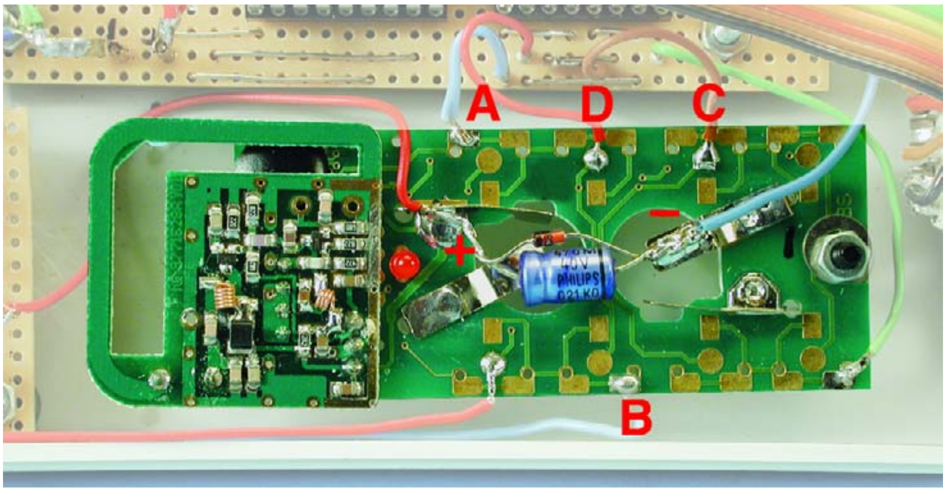

Figure 4. The printed circuit board (author's design) of the handheld transmitter with wires soldered to it to connect to the interface circuit.

Construction

The simplest way to construct the interface circuit is to use matrix stripboard. The components are all widely available and so should be easy to obtain. A suitable latching relay for RE1 is available for example from Conrad Electronics (order code 50 40 60). For RE2 practically any small 12 V relay can be used.

As can be seen from the circuit diagram of the parallel port interface and from the photographs, a few connections need to be made to the remote control transmitter using short lengths of insulated wire. The connection points are easy to find: they are simply the contacts of the pushbuttons. The circuitry inside the transmitter can be quite tightly packed, and so soldering can be a little fiddly. Things are not too difficult with the FS 10 remote control system available from the German company ELV that we used in our prototype: as soon as the enclosure of the transmitter (part code FS 10 S8-2: see Figure 1) is opened up and the caps of the pushbuttons removed, the contact points of the pushbuttons are visible (Figure 4). It is very easy to solder wires to these contacts, connecting the transmitter to the optocouplers (points A to D in the circuit). The pushbuttons can no longer be used.

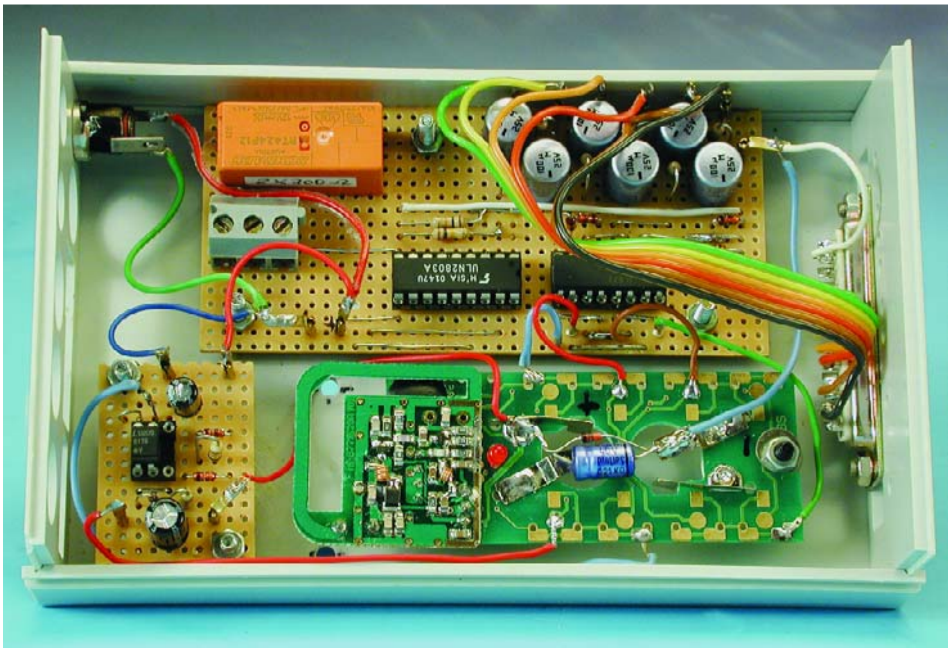

Figure 5. The switching interface built into a plastic enclosure.

The completed assembly can be protected by a plastic enclosure (**Figure 5**, which shows a slightly modified version of the interface).

The Software

The control software is as straightforward as the interface circuit. All it has to do is set the individual bits of the parallel port high for a period of about one second, and then low again. We do this using a COM file, which is created in advance with the aid of a VB (Visual BASIC) script. First type in the program Mk_COM.vbs shown in the text box using any suitable text editor (such as WordPad or Notepad), and save it under this name in any directory on the hard disk. Now run the program by double-clicking on the filename: this will create the file SETLPT.COM in the same directory. The VB script can now be deleted.

The various switching actions can be carried out using the following commands:

SETLPT 1:01 switches the latching relay on.
SETLPT 1:02 switches the latching relay off.
SETLPT 1:08 switches the non-latching relay (briefly) on.
SETLPT 1:10 switches remotely-controlled mains socket 2 on.
SETLPT 1:20 switches remotely-controlled mains socket 2 off.
SETLPT 1:40 switches remotely-controlled mains socket 3 on.
SETLPT 1:80 switches remotely-controlled mains socket 3 off.

The space after 'SETLPT' must not be omitted. The digit '1' after 'SETLPT' and the space refer to parallel port LPT1. The characters after the colon are hexadecimal digits which define the value which will be sent to the parallel port: two digits must always be specified. The assembler routine sets the parallel port to the value given after the colon for about 1 s, and then sets it back to zero again so that only a brief positive pulse will appear on the data line. If you have a second parallel port ('LPT2') on your computer which you wish to use with the interface, then the command should read ‘SETLPT 2:XX’. A note for PC experts: the I/O base address used for LPT1 is 0378 hex, that for LPT2 is 0278 hex.

You can create a shortcut to the file SETLPT.COM with parameter ‘1:08’ on the desktop, and call it `Open Door’. Then, when someone rings the doorbell while you are busy at the computer, just double click on the icon and the door opener will be activated. To prevent the DOS window appearing, right-click on the icon and select ‘Properties’. Then, under ‘Shortcut’ select ‘Run minimised’ and ‘Close window on exit’.

The following file, ‘**Mk_COM.vbs**’, can be typed in and run to create the assembler file ‘**SETLPT.COM**’ automatically. Alternatively, the file ‘**SETLPT.COM**’ can be downloaded directly from the *Elektor Electronics* website at www.elektor-electronics.co.uk (select this issue).

```
    d = “186,120,003,138,030,130,000,128,251,049,”
d = d & “116,007,128,251,050,117,030,254,206,160,”
d = d & “132,000,044,048,177,004,210,224,138,038,”
d = d & “133,000,128,236,048,008,224,177,001,056,”
d = d & “200,116,009,208,225,117,248,184,001,076,”
d = d & “205,033,238,187,000,004,226,254,075,117,”
d = d & “251,184,000,076,238,205,033”
t = split(d, “,”)
Set fso = CreateObject(“Scripting.FileSystemObject”)
Set File = fso.CreateTextFile(“SETLPT.COM”, true)
for n = 0 to ubound(t) : File.write chr(t(n)) : next
```

Conclusion

The author has been using the circuit for many months without any problems, mainly in conjunction with an SMS remote control circuit previously published in *Elektor Electronics*.

If you prefer to control the interfaces using Windows, there are small programs available on the Internet which allow the parallel port signals to be set high or low. Simple parallel port interfaces using Windows software have also been published in the past in *Elektor Electronics*, and that software can be adapted for the interface presented here. Examples include the ‘Universal Parallel Input/Output for PCs’ on page 20 of the April 2000 issue and ‘Centronics Interface’ on page 28 of the April 1996 issue. The book ‘PC Interfaces under Windows’ is also recommended.

(030205-1)